**Test Object Oriented**

1. Due oggetti (istanze della stessa classe) possono avere attributi con lo stesso valore?

* Si. √
* No

1. Specificare vari approcci nel mondo di SW Commerciale.

|  |
| --- |
| Ci sono diverse tipologie di approccio allo sviluppo di software. Alcuni di questi modelli sono: |
| Il modello a cascata, il modello evolutivo, il modello a spirale o metodologie agili come ad esempio |
| SCRUM o l’ extreme programming |

1. Specificare la fabbricazione di un’auto con un approccio Object Oriented.

|  |
| --- |
| Si parte dalla definizione della superclasse Auto, da cui poi derivano tutte le sottoclassi (ovvero i |
| pezzi dell’ auto). I componenti (ad esempio: ruota, motore, fanali ecc..) estendono la superclasse |
| Auto e da essa ereditano gli attributi più generici (ad esempio la targa, il telaio dell’ auto) e invece definiscono quelli specifici (es: motore definirà la cilindrata, fanali definirà il tipo utilizzato). |

1. Che cosa e un diagramma UML?

|  |
| --- |
| UML è un linguaggio di modellazione usato per descrivere le caratteristiche di un sistema. Un |
| Diagramma UML è quindi utile perché appunto modella gli oggetti e ne mostra i nomi delle classi, |
| Gli attributi e i metodi. Serve per poter facilitare poi lo sviluppo degli oggetti stessi |

1. È necessario distruggere gli oggetti creati se non sono utilizzati?

* Si
* No. √

1. Com’è gestita la area di memoria per due oggetti che hanno gli stessi attributi?

|  |
| --- |
| Sono allocate due aree di memoria diverse pur avendo entrambe gli stessi attributi. Ci sarà un area |
| Di memoria dedicata ad auto1 ed una separata dedicata ad auto2 |
|  |

Es:

|  |  |
| --- | --- |
| auto1 | auto2 |
| Marca: Fiat | Marca: Fiat |
| Modello: Punto | Modello: Punto |

1. Come è possibile accedere ad un oggetto?

|  |
| --- |
| È possibile accedere agli attributi di un oggetto utilizzando dei metodi precedentemente creati di |
| get e set. Servono appunto per farsi restituire gli attributi dell’ oggetto. In questo caso |
| auto1.getMarca() e auto2.getMarca() restituiscono entrambe Fiat. I metodi set servono invece per  Definire gli attributi, ad esempio auto1.setModello(Punto). |

1. Il seguente codice permette di accedere ad una funzione della classe Albero?

Albero albero;

albero. setNomeAlbero;

* Si
* No. √

|  |
| --- |
| Un costruttore è un metodo che viene utilizzato per appunto “costruire” ed instanziare un oggetto. |
| Avendolo precedentemente definito un esempio è Auto = new Auto(“Fiat”, “Punto”) assegna alla |
| Marca “Fiat” e al modello “Punto” costruendo quindi l’ oggetto |

1. Che cos’ è un costruttore?
2. Data l’istruzione di seguito, distinguere fra classe, variabile, costruttore

**Fiore rosa = new Fiore ();**

Classe Fiore

Variabile rosa

Costruttore Fiore()

1. Due o più variabili possono puntare allo stesso oggetto?

* Si. √
* No

1. Che Cosa è il garbage collector e come si invoca?

|  |
| --- |
| Il garbage collector lavora automaticamente per gestire la memoria e quindi per allocare / |
| deallocare memoria. La memoria viene allocata ad esempio quando viene chiamato il costruttore |
| Con l’ istruzione “new” e deallocata automaticamente quando non è più accessibile attraverso nessuna variabile. |

1. Quali classi possano accedere ad un package?

Tutte le classi possono accedere ad un package, invece ci sono differenze nell’ accesso alle classi, che dipende appunto dal loro modificatore di visibilità (dal meno stringente public al più stringente protected)

public XXX class.

SI

NO

public final XXX class.

NO

SI

private XXX class.

NO

SI

protected XXX class.

NO

SI

1. Se bisogna invocare uno o più metodi o funzioni di una classe in un’altra classe di cui anche il package è diverso, si può accedere alle classi senza importare il package relativo alla classe stessa?

|  |
| --- |
| Se in un file a.java, si vuole usare una classe c definita in un package b, ci si può riferire |
| ad essa usando b.c, se è definita in un package senza nome allora semplicemente c. Oppure |
| Si può appunto importare il package con import b.\* |

1. Descrivere cosa vuole dire il codice seguente

import package.class1 importa solo la classe del package specificata

import package.\* importa tutte le classe del package

1. è possibile estendere una classe final?

* Si
* No. √

1. Quale e la risposta corretta?

Public static final String SYNCLAB = SYNCLAB;

* Attibuto
* Variabile
* Classe
* Costante. √
* Classe final

1. Cos’è una classe interna? Ne possiamo modificare la visibilità?

|  |
| --- |
| Una classe interna è una classe dichiarata all’ interno di un altra classe. Una classe interna può |
| Fare riferimento a metodi e variabili della classe che la contiene, ma non può contenere variabili o |
| Metodi statici. Si può cambiare la visibilità ma deve essere o uguale o più stringente della classe contenitore |